

[SQUEAKING]

[RUSTLING]

[CLICKING]

**HELENA**  
**VAILLICROSA:** In this section, I'm going to dive into stack and bricks, which are, I would say, books of maps. So just for you to have a good understanding of how stack and bricks work, it's like if you would have each of them map being a piece of paper and you would like to put them together in a book form, then you would use stack and brick.

And how is this helpful? This is helpful because these standardized processes that we want to proceed on the maps. So let's say we want to chop them in the same shape, or we want to transform them in the same resolution, or we want to calculate some results of these maps. We can do it using stack and bricks. And we can also store stack and bricks in stack and brick form in our computer.

So first of all, when we want to create stacks and bricks, there's a very handy way to charge these files all at once, so we don't have to type the route every time. In this case, I'm going to use, as example, the WorldClim data that we downloaded. If you've paid attention inside the folder, it comes with 12 different maps that are representing the average temperature of each month.

And now, we are going to combine them into one brick, so we can calculate the mean or the standard deviation of all of them. So first of all, we need to write the directory route where we have all the maps. Also, you could add in your code a chunk of code saying pattern.tif, has in case, you would have some other files in that folder and you only wanted to upload the TIF. But in this case, everything is TIF. So I don't need to do that.

You charge the map files, which is only the name of the object that's inside. And then we want to transform all these routes into different rasters. And to do so, we are going to use the lapply function here using these map files and transforming them all into raster.

So doing this, we'll get a list of these 12 maps. And we want to just to consult one of them. I'm just going to see what we have in the number 6, for example. And here, we have a map. This is all the information about this map - so the dimensions, resolution, extent, and so forth. And just to see how it looks like, it's just a map.

And we want to include all these maps into a stack or a brick. They are mostly the same for working purposes. But the organization of the data is slightly different. So sometimes, for some processes, you will need to use stack or brick. But the concept behind it is mostly the same. So it works either way, just using the function stack or brick. And they have to be in the same extension and resolution. Otherwise, R wouldn't let you to put them together.

So here we have it. You see we've created a raster brick or raster stack, depending on the function that we used. And now, I'm going to use the function calc, which is a calculator. And I'm going to calculate something on the maps that we just uploaded. In this case, I'm calculating the standard deviation. But I could change this for the mean, for example.

So I'm going to calculate inside this stack right here the standard deviation of all the maps. There is also extra code that you can see here. And this code is used to calculate the time that we are investing in processing this information. Just to make it fair, we have to run it all at once.

It's going to run it. And this is going to extract the system time that was stored in our computer at the beginning of the process. And it's going to record it again at the end of the process. And then it's going to make the calculation of how much time difference there is in between the initiation and the finalization of the process.

So in this case, we've calculated the standard deviation. That took us 14 seconds. And let's see how it looks like. So this is the standard deviation of the average temperatures at global scale. And that's also something we can store in our computer. We can write raster, as it was only one raster. So for stacks and bricks, it works the same.

So instead of having one map here, we just could have put the raster brick or stack. In this case, I'm just going to store the outcome of this calculation that we've made and just put the route that you want the things to be stored on. And if you run this code, you're going to create this map in your computer.

And also, another way to store the map without using this function is to-- so while you are calculating something where you are running a process in a raster function, then if you specify a file name, R is going to understand that while this is being processed, it's going to be at the same time stored. So only with a line of code, you would have the creation of that map plus the saving process in your computer.

Also, in case you would like to store stack or brick, you would have to change from TIF extension to GRD. When you create this extension, you would have two different files. And these two different files are-- well, it's necessary that both would be in the same folder. Otherwise, when you would read them, R would understand them as corrupted. So make sure that you're not separating these two files in order to be able to read the code appropriately.

Also, moving on to the next function, we're going to move to the crop function. This can be used to crop one map. Or it can be used to crop a brick or a stack. And chopping them in small pieces, in this case, I'm going to use as a base this temperature map. And I'm just going to chop the latitude and longitude based on the Eastern USA Coast.

So if I do this, our temperature is not found. I should go and charge it again. Here we have it. Now, I should be able to crop it. I want to see how it looks like. And this is the result of a cropping. This is a very handy tool in order to make experiments with your code.

So as you have seen before, sometimes doing some processes when we are working in maps, it can take a long time. So if you want to just check that your code is working and that all the processes that has been done are what you want it to be, you can just cut your map in a small piece, and then run everything in that small piece, just to be a sample of your final code. And then when you've made sure that everything is running properly, you can run the process with the whole map. So that might save you some time.

There is also another function that I particularly think is very useful, which is the mask function. Here, I'm going to use one of the maps that we've used previously up in the code, which was this nitrogen deposition map. Just to remind you what was that, I'm just going to charge it and plot it.

So this is the nitrogen deposition that has happened in 1901, I think. But we see that the map here, it covers the whole surface of the Earth, including water bodies. What happens if I want to have the information only on the ground, only on the Earth's surface?

So what I'm going to do is I'm going to mask this map right here with the maps that we previously used that included only-- that I excluded water bodies, let's say. So I'm going to use again this temperature map. I'm just going to plot it. I'm going to use it as a base.

So to do so there, is this function right here that's called mask. And I'm going to mask this nitrogen deposition based on this shape right here. I'm going to run it, it's going to give me an error. Here you have it. This error is saying that we have different extent. What is that? Let's just check the two maps.

This map and this other map, it's complaining about the extent. So this extent here is this one. And this extent here is slightly different. And that's why we have this problem. How to solve it? Well, we'll need to fix this extent problem. We're going to use this resample function here that's going to transform both maps in the same extension, projection, and resolution all at once.

So by using this, I'm going to make the nitrogen deposition map to have the same features as the other map. So in this case, I'm going to use the second map that we produced to see if it works. And now, the error is solved. And we can see the result on how the map has changed and how all the water bodies has been masked.